Algorithms for Leader Selection in Stochastically Forced Consensus Networks

Fu Lin, Makan Fardad, and Mihailo R. Jovanović

Abstract

We examine the leader selection problem in stochastically forced consensus networks. A node is a leader if, in addition to relative information from its neighbors, it also has an access to its own state. This problem arises in several applications including control of vehicular formations and localization in sensor networks. We are interested in selecting an *a priori* specified number of leaders such that the steady-state variance of the deviation from consensus is minimized. Even though we establish convexity of the objective function, combinatorial nature of constraints makes determination of the global minimum difficult for large networks. We introduce a convex relaxation of constraints to obtain a lower bound on the global optimal value. We also use a simple but efficient greedy algorithm and the alternating direction method of multipliers to compute upper bounds. Furthermore, for networks with noise-free leaders that perfectly follow their desired trajectories, a sequence of convex relaxations is used to identify the leaders. Several examples ranging from regular lattices to random graphs are provided to illustrate the effectiveness of the developed algorithms.

Index Terms

Alternating direction method of multipliers, consensus networks, convex optimization, convex relaxations, greedy algorithm, leader selection, performance bounds, semidefinite programming, sensor selection, sparsity, variance amplification.

I. Introduction

Reaching consensus in a decentralized fashion is an important problem in network science [1]. This problem is often encountered in social networks where a group of individuals is trying to agree on a certain issue [2], [3]. A related problem has been studied extensively in computer

Financial support from the National Science Foundation under CAREER Award CMMI-06-44793 and under awards CMMI-09-27720 and CMMI-0927509 is gratefully acknowledged.

F. Lin and M. R. Jovanović are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455. M. Fardad is with the Department of Electrical Engineering and Computer Science, Syracuse University, NY 13244. E-mails: fu@umn.edu, makan@syr.edu, mihailo@umn.edu.

science with the objective of distributing evenly computational load over a network of processors [4], [5]. Recently, consensus problem has received considerable attention in the context of distributed control [6]–[9]. For example, in cooperative control of vehicular formations, it is desired to use local interactions between vehicles in order to reach agreement on quantities such as heading angle, velocity, and inter-vehicular spacing. Since vehicles have to maintain agreement in the presence of uncertainty it is of importance to study robustness of consensus. Several authors have recently used the steady-state variance of the deviation from consensus to characterize performance limitations of stochastically forced networks [10]–[16].

In this paper, we consider undirected consensus networks with two groups of nodes. Ordinary nodes, so-called *followers*, form their action using relative information exchange with their neighbors; special nodes, so-called *leaders*, also have access to their own states. This setting may arise in the control of vehicular formations where all vehicles are equipped with ranging devices (that provide information about relative distances with respect to their neighbors), and the leaders additionally have GPS devices (that provide information with respect to a global frame of reference).

We are interested in assigning an *a priori* specified number of nodes as leaders in order to minimize the steady-state variance of the deviation from consensus. For undirected networks in which all nodes are subject to stochastic disturbances, we establish convexity of the objective function. In spite of this, combinatorial nature of Boolean constraints (a node is either a leader or it is not) makes determination of the global minimum challenging for large networks. Instead, we focus on computing lower and upper bounds on the global optimal value. Convex relaxation of Boolean constraints is used to obtain a lower bound, and two different algorithms are used to obtain an upper bound and to identify leaders. The first algorithm utilizes one-leader-at-a-time (greedy) approach followed by a swap procedure that improves performance by checking possible swaps between leaders and followers. In both steps, algorithmic complexity is significantly reduced by exploiting structure of low-rank modifications to Laplacian matrices. The second algorithm utilizes the alternating direction method of multipliers (ADMM) [17] which is capable of handling the nonconvex Boolean constraints by a simple projection. Computational efficiency of these algorithms makes them well-suited for establishing achievable performance bounds for leader selection problem in large stochastically forced networks.

Following [18]–[21], we also examine consensus networks in which leaders follow desired

trajectories at all times. For this idealized case, the identification of *noise-free leaders* is additionally complicated by nonconvexity of the objective function. For consensus networks with at least one leader, adding leaders always improves performance [18]. In view of this, a greedy algorithm that selects one leader at a time by assigning the node that leads to the largest performance improvement as a leader was proposed in [18]. Furthermore, it was proved in [20] that the variance of deviation from consensus is a supermodular function of the set of noise-free leaders. Thus, the supermodular optimization framework in conjunction with the greedy algorithm can be used to provide selection of leaders that is within a provable bound from globally optimal solution [20].

In contrast to the above references, we use convex optimization to select noise-free leaders. An alternative explicit expression for the objective function that we provide is used to identify the source of nonconvexity and to suggest an LMI-based convex relaxation. In addition to this, we relax the *hard* Boolean-valued constraint on the number of leaders with a *soft* one. This is achieved by augmenting the objective function with the sparsity-promoting term that penalizes the ℓ_1 norm of the vector of optimization variables [22], [23]. The ℓ_1 norm provides a means for obtaining a sparse solution whose nonzero elements identify the leaders. The developed algorithm produces a tradeoff curve between the number of noise-free leaders and the variance of the deviation from consensus by solving a parameterized family of convex optimization problems.

The controllability of leader-follower consensus networks is also an active area of research [24]–[27]. Recent efforts have focused on characterizing graph-theoretic conditions for controllability of the network in which a number of pre-specified leaders act as control inputs. In contrast, the leader selection problem aims at identifying leaders that are most effective in maintaining consensus in the presence of disturbances. Other related work on augmenting topologies of networks to improve algebraic connectivity includes [28]–[30].

The paper is organized as follows. In Section II, we formulate the leader selection problem and establish connections with the sensor selection problem. In Section III, we propose an LMI-based convex relaxation of the objective function in the noise-free leader selection problem. Furthermore, instead of imposing Boolean constraints, we augment the objective function with the ℓ_1 norm of the optimization variable. In Section IV, we develop efficient algorithms to compute lower and upper bounds on the global optimal value for the noise-corrupted leader selection problem. Finally, we conclude the paper with a summary of our contributions in Section V.

II. PROBLEM FORMULATION

In this section, we formulate the noise-corrupted and noise-free leader selection problems in consensus networks and make connections to sensor selection problem in sensor networks. Furthermore, we establish an equivalence between the noise-corrupted and noise-free leader selection problems when all leaders use arbitrarily large feedback gains on their own states.

A. Leader selection problem in consensus networks

We consider n single-integrators

$$\dot{\psi}_i \,=\, u_i \,+\, w_i, \quad i \,=\, 1, \ldots, n$$

where ψ_i is the scalar state, u_i is the control input, and w_i is the white stochastic disturbance with zero-mean and unit-variance. A node is a *follower* if it uses *only* relative information exchange with its neighbors to form its control action,

$$u_i \,=\, -\sum_{j\,\in\,\mathcal{N}_i} (\psi_i - \psi_j).$$

A node is a *leader* if, in addition to relative information exchange with its neighbors, it also has access to its own state

$$u_i \,=\, -\sum_{j\,\in\,\mathcal{N}_i} (\psi_i - \psi_j) \,-\, \kappa_i\, \psi_i.$$

Here, κ_i is a positive number and $\mathcal{N}_i$ is the set of all nodes that node i communicates with.

The communication network is modeled by a connected, undirected graph; thus, the graph Laplacian L is a symmetric positive semidefinite matrix with a single eigenvalue at zero and the corresponding eigenvector $\mathbb{1}$ of all ones. A state-space representation of the leader-follower consensus network is given by

$$\dot{\psi} \,=\, -\,(L \,+\, D_\kappa D_x)\,\psi \,+\, w \tag{1}$$

where

$$D_\kappa \,:=\, \mathrm{diag}\,(\kappa)\,, \quad D_x \,:=\, \mathrm{diag}\,(x)$$

are diagonal matrices formed from the vectors $\kappa = [\,\kappa_1 \;\cdots\; \kappa_n\,]^T$ and $x = [\,x_1 \;\cdots\; x_n\,]^T$. Here, x is a Boolean-valued vector with its ith entry $x_i \in \{0,\, 1\}$, indicating that node i is a leader if

$x_i = 1$ and that node i is a follower if $x_i = 0$. In connected networks with at least one leader $L + D_\kappa D_x$ is a positive definite matrix and the steady-state covariance of ψ is determined by

$$\begin{aligned}\Sigma &:= \lim_{t\to\infty} \mathcal{E}\left(\psi(t)\,\psi^T(t)\right)\\ &= \int_0^\infty \mathrm{e}^{-(L+D_\kappa D_x)t}\,\mathrm{e}^{-(L+D_\kappa D_x)^T t}\,\mathrm{d}t\\ &= \frac{1}{2}\,(L+D_\kappa D_x)^{-1}.\end{aligned}$$

Following [12], [15], we use the total steady-state variance

$$\mathrm{trace}\,(\Sigma) \;=\; \frac{1}{2}\,\mathrm{trace}\left((L+D_\kappa D_x)^{-1}\right) \tag{2}$$

to quantify performance of consensus networks subject to stochastic disturbances.

We are interested in identifying N_l leaders that are most effective in reducing the steady-state variance (2). For an *a priori* specified number of leaders $N_l < n$, the leader selection problem can thus be formulated as

$$\begin{array}{lrcl}\underset{x}{\text{minimize}} & J(x) & = & \mathrm{trace}\,((L\,+\,D_\kappa D_x)^{-1})\\ \text{subject to} & x_i & \in & \{0,1\}, \qquad i\,=\,1,\ldots,n\\ & \mathbb{1}^T x & = & N_l.\end{array} \tag{LS1}$$

In (LS1), the number of leaders N_l as well as the matrices L and D_κ are the problem data, and the vector x is the optimization variable. As shown in Section IV, for a positive definite matrix $L + D_\kappa D_x$, the objective function J in (LS1) is a convex function of x. The challenging aspect of (LS1) comes from the nonconvex Boolean constraints $x_i \in \{0,1\}$; in general, finding the solution to (LS1) requires an intractable combinatorial search.

Since the leaders are subject to stochastic disturbances, we refer to (LS1) as the *noise-corrupted* leader selection problem. We also consider the selection of *noise-free* leaders which follow their desired trajectories at all times [18]. Equivalently, in coordinates that determine deviation from the desired trajectory, the state of every leader is identically equal to zero, thereby yielding only the dynamics of the followers

$$\dot{\psi}_f \;=\; -\,L_f\,\psi_f \;+\; w_f.$$

Here, L_f is obtained from L by eliminating all rows and columns associated with the leaders.

Thus, the problem of selecting leaders that minimize the steady-state variance of ψ_f amounts to

$$\begin{array}{lrcl} \underset{x}{\text{minimize}} & J_f(x) & = & \text{trace}\,(L_f^{-1}) \\ \text{subject to} & x_i & \in & \{0,1\}, \qquad i \,=\, 1,\ldots,n \\ & \mathbb{1}^T x & = & N_l. \end{array} \tag{LS2}$$

As in (LS1), the Boolean constraints $x_i \in \{0,1\}$ are nonconvex. Furthermore, as we demonstrate in Section III, the objective function J_f in (LS2) is a nonconvex function of x.

In what follows, we establish equivalence between the noise-corrupted and noise-free leader selection problems (LS1) and (LS2) when all leaders use arbitrarily large feedback gains on their own states. Partitioning ψ into the state of the leader nodes ψ_l and the state of the follower nodes ψ_f brings system (1) to the following form[1]

$$\left[\begin{array}{c} \dot{\psi}_l \\ \dot{\psi}_f \end{array}\right] \,=\, -\left[\begin{array}{cc} L_l + D_{\kappa_l} & L_0 \\ L_0^T & L_f \end{array}\right]\left[\begin{array}{c} \psi_l \\ \psi_f \end{array}\right] + \left[\begin{array}{c} w_l \\ w_f \end{array}\right]. \tag{3}$$

Here, $D_{\kappa_l} := \text{diag}\,(\kappa_l)$ with $\kappa_l \in \mathbb{R}^{N_l}$ being the vector of feedback gains associated with the leaders. Taking the trace of the inverse of the 2×2 block matrix in (3) yields

$$J \,=\, \text{trace}\left(L_f^{-1} \,+\, L_f^{-1}\, L_0^T\, S_{\kappa_l}^{-1}\, L_0\, L_f^{-1} \,+\, S_{\kappa_l}^{-1}\right)$$

where

$$S_{\kappa_l} \,=\, L_l \,+\, D_{\kappa_l} \,-\, L_0\, L_f^{-1}\, L_0^T$$

is the Schur complement of L_f. Since $S_{\kappa_l}^{-1}$ vanishes as each component of the vector κ_l goes to infinity, the variance of the network is solely determined by the variance of the followers, $J_f = \text{trace}\left(L_f^{-1}\right)$, where L_f is the reduced Laplacian matrix obtained by removing all columns and rows corresponding to the leaders from L.

[1]Note that D_x does not show in (3) since the partition is performed with respect to the indices of the 0 and 1 diagonal elements of D_x.

B. Connections to the sensor selection problem

The problem of estimating a vector $\psi \in \mathbb{R}^n$ from m relative measurements corrupted by additive white noise

$$y_{ij} \;=\; \psi_i \;-\; \psi_j \;+\; w_{ij}$$

arises in distributed localization in sensor networks. We consider the simplest scenario in which all ψ_i's are scalar-valued, with ψ_i denoting the position of sensor i; see [10], [11] for vector-valued localization problems. Let $\mathcal{I}_r$ denote the index set of the m pairs of distinct nodes between which the relative measurements are taken and let e_{ij} belong to $\mathbb{R}^n$ with 1 and -1 at its ith and jth elements, respectively, and zero everywhere else. Then,

$$y_{ij} \;=\; e_{ij}^T\,\psi \;+\; w_{ij}, \quad (i,j) \;\in\; \mathcal{I}_r$$

or, equivalently in the matrix form,

$$y_r \;=\; E_r^T\psi \;+\; w_r \tag{4}$$

where y_r is the vector of relative measurements and $E_r \in \mathbb{R}^{n\times m}$ is the matrix whose columns are determined by e_{ij} for $(i,j) \in \mathcal{I}_r$. Since $\psi + a\mathbb{1}$ for any scalar a results in the same y_r, with relative measurements the position vector ψ can be determined only up to an additive constant. This can also be verified by noting that $E_r^T\mathbb{1} = 0$.

Suppose that N_l sensors can be equipped with GPS devices that allow them to measure their absolute positions

$$y_a \;=\; E_a^T\psi \;+\; E_a^T w_a$$

where $E_a \in \mathbb{R}^{n\times N_l}$ is the matrix whose columns are determined by e_i, the ith unit vector in $\mathbb{R}^n$, for $i \in \mathcal{I}_a$, the index set of absolute measurements. Then the vector of all measurements is given by

$$\begin{bmatrix} y_r \\ y_a \end{bmatrix} \;=\; \begin{bmatrix} E_r^T \\ E_a^T \end{bmatrix} \psi \;+\; \begin{bmatrix} I & 0 \\ 0 & E_a^T \end{bmatrix} \begin{bmatrix} w_r \\ w_a \end{bmatrix} \tag{5}$$

where w_r and w_a are zero-mean white stochastic disturbances with

$$\mathcal{E}(w_r w_r^T) \;=\; W_r, \quad \mathcal{E}(w_a w_a^T) \;=\; W_a, \quad \mathcal{E}(w_r w_a^T) \;=\; 0.$$

In Appendix A, we show that the problem of choosing N_l absolute position measurements among n sensors to minimize the variance of the estimation error is equivalent to the noise-corrupted leader selection problem (LS1). Furthermore, when the positions of N_l sensors are known *a priori* we show that the problem of assigning N_l sensors to minimize the variance of the estimation error amounts to solving the noise-free leader selection problem (LS2).

III. Linear approximation and soft constraint method: noise-free leaders

In this section, we provide an alternative expression for the objective function J_f in the noise-free leader selection problem (LS2). We use this explicit expression to identify the source of nonconvexity and to suggest an LMI-based convex approximation. We then relax the hard constraint of having exactly N_l leaders in (LS2) by augmenting the objective function J_f with the ℓ_1 norm of the optimization variable x. This *soft constraint* approach yields a parameterized family of optimization problems whose solution provides a tradeoff between the ℓ_1 norm of x and the convex approximation of the variance amplification of the network.

A. Explicit expression for the objective function

Since the objective function J_f in (LS2) is not expressed explicitly in terms of the optimization variable x, it is difficult to examine its basic properties (including convexity). We next provide an alternative expression for J_f that allows us to establish lack of convexity and to suggest an LMI-based convex approximation of J_f.

Proposition 1: For networks with at least one leader, the objective function J_f in the noise-free leader selection problem (LS2) can be written as

$$J_f \,=\, \mathrm{trace}\left(L_f^{-1}\right) \,=\, \mathrm{trace}\left((I \,-\, D_x)(G \,+\, D_x \circ L)^{-1}(I \,-\, D_x)\right) \tag{6}$$

where $\circ$ denotes the elementwise multiplication of matrices, and

$$G \,=\, (I \,-\, D_x)\, L\, (I \,-\, D_x), \quad D_x \,=\, \mathrm{diag}\,(x)\,, \quad x_i \,\in\, \{0,1\}, \quad i \,=\, 1,\ldots,n.$$

Furthermore, J_f is a nonconvex function of x over the smallest convex set $x_i \,\in\, [0,1]$ that contains feasible points $x_i \in \{0,1\}$ for $i = 1,\ldots,n$.

Proof: After an appropriate relabeling of the nodes (as done in (3)), L and D_x can be partitioned conformably into 2×2 block matrices,

$$L = \begin{bmatrix} L_l & L_0 \\ L_0^T & L_f \end{bmatrix}, \quad D_x = \begin{bmatrix} I_{N_l \times N_l} & 0_{N_l \times p} \\ 0_{p \times N_l} & 0_{p \times p} \end{bmatrix}, \quad p := n - N_l$$

which leads to

$$G = \begin{bmatrix} 0_{N_l \times N_l} & 0_{N_l \times p} \\ 0_{p \times N_l} & L_f \end{bmatrix}, \quad D_x \circ L = \begin{bmatrix} I_{N_l \times N_l} \circ L_l & 0_{N_l \times p} \\ 0_{p \times N_l} & 0_{p \times p} \end{bmatrix}$$

$$G + D_x \circ L = \begin{bmatrix} I_{N_l \times N_l} \circ L_l & 0_{N_l \times p} \\ 0_{p \times N_l} & L_f \end{bmatrix}.$$

Since $I_{N_l \times N_l} \circ L_l$ is a diagonal matrix with positive diagonal elements and since the principal submatrix L_f of the Laplacian L is positive definite for connected graphs [1, Lemma 10.36], we have

$$G + D_x \circ L \succ 0. \tag{7}$$

Consequently,

$$\operatorname{trace}\left((I - D_x)(G + D_x \circ L)^{-1}(I - D_x)\right) = \operatorname{trace}\left(L_f^{-1}\right)$$

which yields the desired result (6).

We next use a counterexample to illustrate the lack of convexity of J_f over $x_i \in [0, 1]$. Let

$$L = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad D_x = \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix}$$

with $x_1 \in [0, 1]$ and $x_2 = 1$. From

$$G + L \circ D_x = \begin{bmatrix} (1 - x_1)^2 + x_1 & 0 \\ 0 & 1 \end{bmatrix} \succ 0 \quad \text{and} \quad J_f = \frac{(1 - x_1)^2}{(1 - x_1)^2 + x_1}$$

it can be verified that, for $x_1 \in [0, 1/3]$, the second derivative of J_f with respect to x_1 is negative (implying that J_f is not convex). ■

Explicit expression (6) in conjunction with Schur complement can be used to convert the minimization of J_f into the following problem

$$\begin{array}{ll} \underset{X,\,x}{\text{minimize}} & \text{trace}\,(X) \\ \text{subject to} & \left[\begin{array}{cc} X & I\,-\,D_x \\ I\,-\,D_x & G\,+\,D_x\circ L \end{array}\right] \succeq 0 \end{array} \tag{8}$$

where $X \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. To see this, note that since $G+D_x\circ L \succ 0$, we have

$$\left[\begin{array}{cc} X & I-D_x \\ I-D_x & G+D_x\circ L \end{array}\right] \succeq 0 \quad\Leftrightarrow\quad X \succeq (I-D_x)(G+D_x\circ L)^{-1}(I-D_x).$$

Thus, to minimize $\text{trace}\,(X)$ subject to the inequality constraint, we take $X = (I-D_x)(G+D_x\circ L)^{-1}(I-D_x)$, which shows equivalence between the objective functions in (8) and in (6).

Thus, the noise-free leader selection problem (LS2) can be formulated as

$$\begin{array}{ll} \underset{X,\,x}{\text{minimize}} & \text{trace}\,(X) \\ \text{subject to} & \left[\begin{array}{cc} X & I\,-\,D_x \\ I\,-\,D_x & G\,+\,D_x\circ L \end{array}\right] \succeq 0 \\ & G \,=\, (I\,-\,D_x)\,L\,(I\,-\,D_x) \\ & D_x \,=\, \text{diag}\,(x)\,, \quad \mathbb{1}^T x \,=\, N_l, \quad x_i \in \{0,1\}, \quad i \,=\, 1,\ldots,n. \end{array} \tag{9}$$

In addition to the Boolean constraints, the quadratic dependence of G on D_x provides another source of nonconvexity in (9). Thus, in contrast to (LS1), relaxation of the Boolean constraints to $x_i \in [0,1]$ for $i = 1,\ldots,n$ is not enough to guarantee convexity of the optimization problem (9).

B. Linear approximation and soft constraint method

As established in Section III-A, the alternative formulation (9) of the noise-free leader selection problem (LS2) identifies two sources of nonconvexity: the quadratic matrix inequality and the Boolean constraints. In view of this, we use linearization of the matrix G to approximate the quadratic matrix inequality in (9) with an LMI. Furthermore, instead of imposing Boolean constraints, we augment the objective function with the ℓ_1 norm of x. This choice is used as a proxy for obtaining a sparse solution x whose nonzero elements identify the leaders.

The idea of using linearization comes from [31], where a linear approximation of the objective function $\mathrm{trace}\,(YZ)$ at the point (Y_0, Z_0) was considered

$$(1/2)\,\mathrm{trace}\,(Y_0 Z + Y Z_0).$$

To design fixed-order output feedback controllers, the authors of [31] minimize $\mathrm{trace}\,(Y_0 Z + Y Z_0)$ with respect to Y and Z, set $Y_0 \leftarrow Y$, $Z_0 \leftarrow Z$, and repeat. Motivated by this iterative scheme, we consider the following linear approximation of G

$$G_0 \;:=\; (1/2)\,(I \,-\, D_x)\,L\,(I \,-\, D_{x_0}) \,+\, (1/2)\,(I \,-\, D_{x_0})\,L\,(I \,-\, D_x) \tag{10}$$

where D_{x_0} is our current-best-estimate of D_x. Replacing G with G_0 leads to an LMI approximation of the quadratic matrix inequality in (9).

In addition to the linearization, we relax the *hard* constraint $\mathbb{1}^T x = N_l$ for Boolean-valued x with a *soft* one. This is achieved by augmenting the objective function with the ℓ_1 norm of x,

$$\mathrm{trace}\,(X) \,+\, \gamma \sum_{i=1}^{n} |x_i|$$

where the positive number γ characterizes our emphasis on the sparsity of the vector x. We note that the ℓ_1 norm $\|x\|_{\ell_1}$ is a widely used proxy for promoting sparsity [32, Chapter 6]. Putting this soft constraint approach and linearization (10) together, we obtain a *convex* optimization problem

$$\begin{array}{ll} \underset{X,\,x}{\text{minimize}} & \mathrm{trace}\,(X) + \gamma \displaystyle\sum_{i=1}^{n} |x_i| \\ \text{subject to} & \begin{bmatrix} X & I \,-\, D_x \\ I \,-\, D_x & G_0 \,+\, D_x \circ L \end{bmatrix} \;\succeq\; 0 \\ & G_0 \;=\; (1/2)\,(I \,-\, D_x)\,L\,(I \,-\, D_{x_0}) \,+\, (1/2)\,(I \,-\, D_{x_0})\,L\,(I \,-\, D_x) \\ & D_x \;=\; \mathrm{diag}\,(x) \end{array} \tag{11}$$

which can be solved efficiently for small size problems (e.g., $n \leq 30$) using standard software such as CVX [33]. For large problems, we develop a customized algorithm in Appendix B.

For a fixed value of γ, we start with $D_{x_0} = 0$ and solve problem (11) as part of an iterative loop; the solution $D_x = \mathrm{diag}\,(x)$ at every iteration is treated as the current-best-estimate $D_{x_0} =$

$\mathrm{diag}\,(x_0)$ for the linearization in the next iteration until $\|x - x_0\|_2 \leq \epsilon$. Ranging γ from small to large values, the solution to the γ-parameterized family of problems (11) provides a tradeoff between minimization of $\mathrm{trace}\,(X)$ and minimization of $\|x\|_{\ell_1}$. Larger values of γ promote smaller $\|x\|_{\ell_1}$ and *typically* lead to fewer nonzero elements in x. Depending on the structure of the network, there may not exist values of γ that lead to a vector x with exactly N_l nonzero elements. In this case, we find the solution x^* that has the least number of nonzero elements N^* with $N^* > N_l$, and use the indices of the N_l largest entries of x^* to determine the leaders.

C. Examples

1) An example from [18]*:* We next use the soft constraint method of Section III-B to select leaders for a small network with 25 nodes shown in Fig. 1. As shown in Figs. 2a and 2b, the number of leaders N_l decreases and the variance J_f of the followers increases with γ. The tradeoff between the number of leaders and the variance of followers is illustrated in Fig. 2c.

Figure 3 compares performance of the soft constraint method to performance of the greedy algorithm [18]–[20], which chooses one leader at a time by assigning the node that provides the largest performance improvement as a leader. Using a supermodular optimization framework, it was shown in [20] that the greedy algorithm selects noise-free leaders that are within a provable performance bound from the global solution to (LS2). This motivates use of greedy algorithm as a benchmark for performance of the soft constraint method. As shown in Fig. 3a, for small number of leaders (e.g., $N_l \leq 5$), the greedy algorithm outperforms the soft constraint method; the only exception happens for $N_l = 3$. A more detailed comparison is reported in Table I, with the global solution to (LS2) for $N_l \leq 5$ resulting from exhaustive search.

When the number of leaders is large (e.g., $N_l \geq 9$), the soft constraint method outperforms the greedy algorithm; see Fig. 3b. The heuristics of assigning nodes with large degrees (i.e., large number of neighbors) as leaders is outperformed by both greedy and soft constraint methods. The poor performance of the simple degree-heuristics-based-selection was also noted in [18]–[20].

2) A random network example: We next consider the selection of noise-free leaders in a network with 100 randomly distributed nodes in a unit square. A pair of nodes can communicate with each other if their distance is not greater than 0.2. This scenario arises in sensor networks with prescribed omnidirectional (i.e., disk shape) sensing range [1]. As shown in Figs. 4a and 4b,

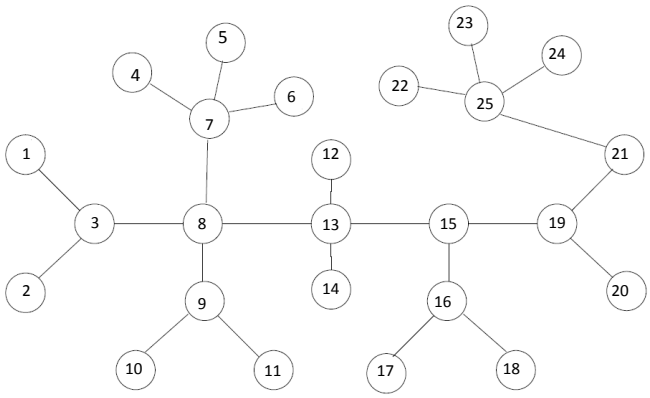
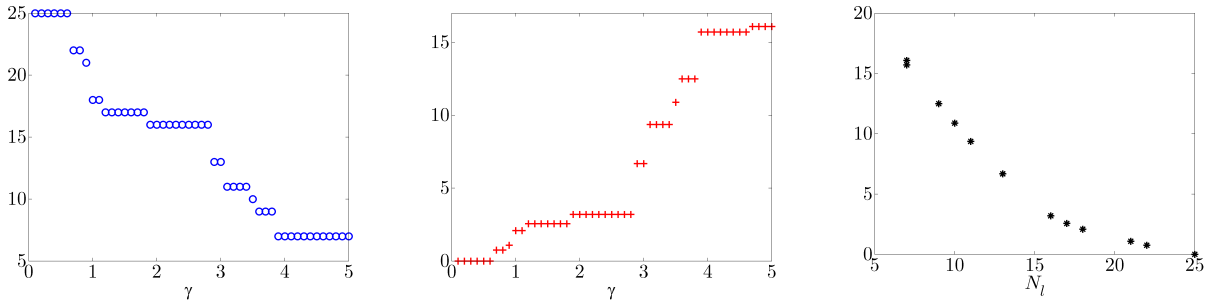
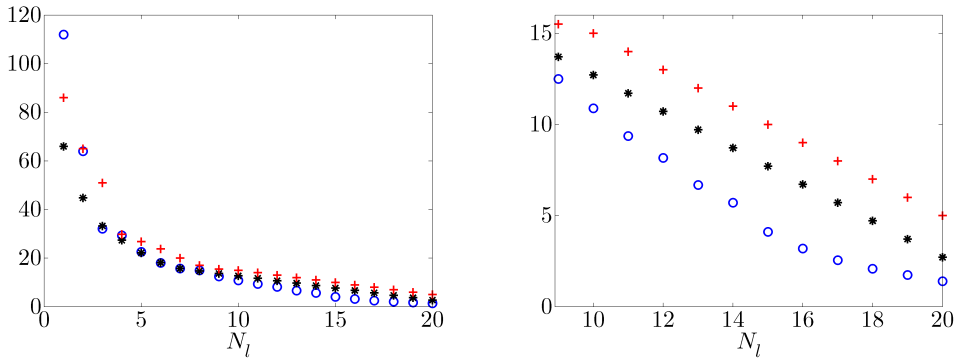


Fig. 1: A small network with 25 nodes [18].



(a) Number of leaders N_l (b) Variance of the network J_f (c) Tradeoff between N_l and J_f

Fig. 2: Performance of the soft constraint method for the network shown in Fig. 1: (a) the number of leaders N_l decreases with γ; (b) the variance of the followers J_f increases with γ; and (c) the tradeoff between N_l and J_f.



(a) (b)

Fig. 3: (a) The variance of the followers J_f obtained using the soft constraint method ($\circ$), the greedy algorithm ($*$), and the degree heuristics ($+$) for the network shown in Fig. 1. (b) Comparison of three algorithms for $N_l \geq 9$.

TABLE I: Performance comparison of greedy algorithm and soft constraint method with the global solution to the noise-free leader selection problem (LS2) for the network shown in Fig. 1.

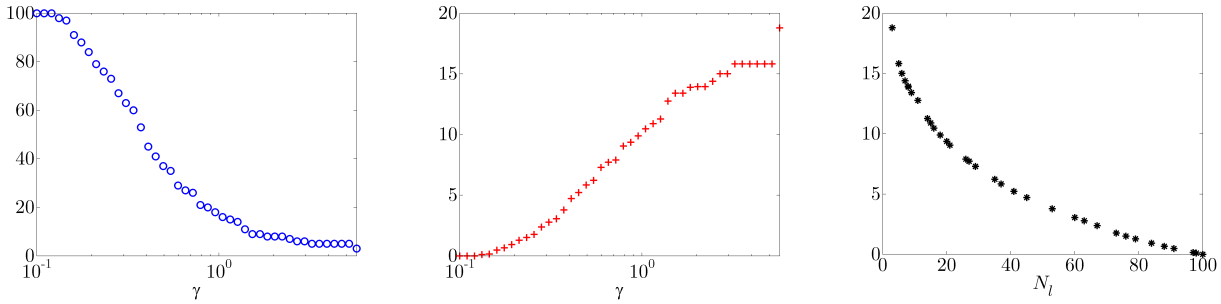
	global solution		greedy algorithm		soft constraint	
N_l	J_f	leaders	J_f	leaders	J_f	leaders
1	66.0	13	66.0	13	112.0	25
2	38.4	8, 25	44.8	13, 25	64.0	16, 25
3	30.0	8, 16, 25	33.3	7, 13, 25	32.1	7, 16, 25
4	25.3	7, 9, 16, 25	27.4	7, 13, 16, 25	29.4	7, 16, 20, 25
5	20.7	3, 7, 9, 16, 25	22.2	3, 7, 13, 16, 25	22.6	3, 7, 16, 20, 25

the number of leaders N_l decreases and the variance J_f of followers increases with γ; also see the tradeoff curve between N_l and J_f in Fig. 4c.

For this random network example, we observe similar selection of leaders and similar performance of the soft constraint and greedy algorithms. Furthermore, for $N_l > 1$, both these algorithms significantly outperform the degree-heuristics-based-selection; see Fig. 5. To gain some insight into the selection of leaders, we compare the results obtained using soft constraint method and the degree heuristics. As shown in Fig. 6b, the degree heuristics chooses nodes that turn out to be in the proximity of each other. In contrast, the soft constraint method select leaders that, in addition to having large degrees, are far from each other; see Fig. 6a. As a result, the selected leaders can influence more followers and thus more effectively improve the performance of the network.

The contrast between degree heuristics and soft constraint method becomes even more dramatic for large number of leaders. As shown in Figs. 6c and 6d, the leader sets obtained using the soft constraint method and degree heuristics are almost *complements* of each other. While the degree heuristics clusters the leaders around the center of the network, the soft constraint method distributes the leaders around the boundary of the network.

Figures 7a and 7b show the degree distribution of all the nodes in the random network and of the 41 nodes that are selected as leaders (see Fig. 6c). In contrast to the degree heuristics, the soft constraint method chooses nodes with both large- and small-degrees as leaders; in particular, all nodes with degree less than 8 and all nodes with degree greater than 18 are selected.



(a) Number of leaders N_l (b) Variance of the network J_f (c) Tradeoff curve between N_l and J_f

Fig. 4: A random network with 100 nodes: (a) the number of leaders N_l decreases with γ; (b) the variance of the followers J_f increases with γ; and (c) the tradeoff curve between N_l and J_f.

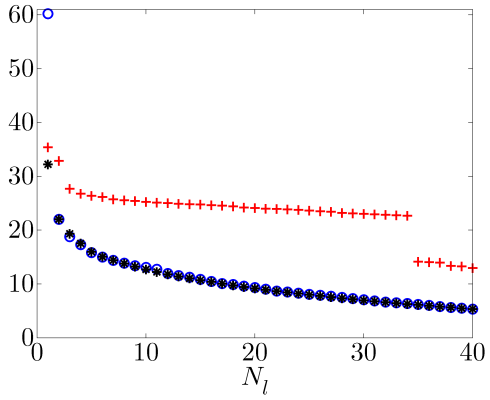
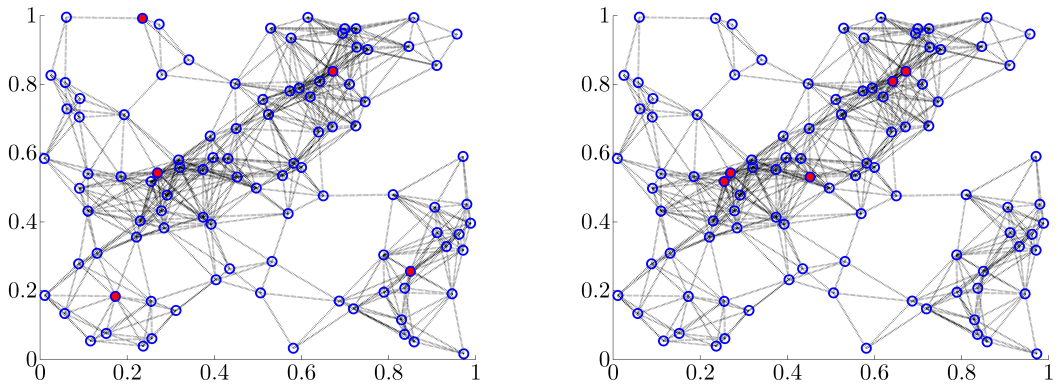


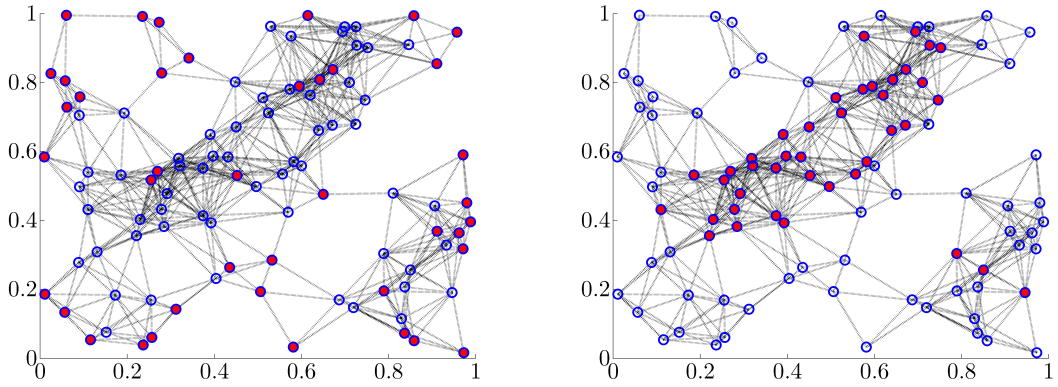
Fig. 5: The objective function J_f obtained using the soft constraint method ($\circ$), the greedy algorithm ($*$), and the degree heuristics ($+$) for the random network.

IV. Lower and upper bounds on global performance: noise-corrupted leaders

In contrast to the noise-free leader selection problem (LS2), we next show that the objective function in the noise-corrupted leader selection problem (LS1) is convex. We take advantage of the convexity of J in (LS1) and develop efficient algorithms to compute lower and upper bounds on the global optimal value $J_{\rm opt}$ of (LS1). A lower bound results from convex relaxation of Boolean constraints in (LS1). Furthermore, upper bounds are obtained using an efficient greedy algorithm and the alternating direction method of multipliers (ADMM). Greedy algorithm selects one leader at a time, which introduces low-rank modifications to the Laplacian matrix. We exploit this feature in conjunction with the matrix inversion lemma to gain computational efficiency. On the other hand, the ADMM algorithm handles the Boolean constraints explicitly by a simple



(a) $N_l = 5$ (b) $N_l = 5$



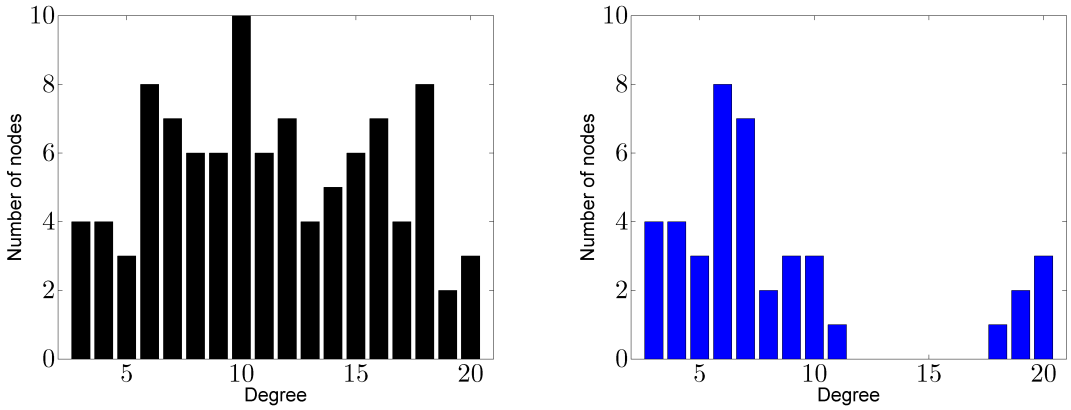
(c) $N_l = 41$ (d) $N_l = 40$

Fig. 6: Selection of leaders (●) for the random network example using soft constraint method in (a) and (c) and using degree heuristics in (b) and (d).

projection onto a discrete nonconvex set. Finally, we provide two examples to illustrate the performance of the developed approach.

A. *Convex relaxation to obtain a lower bound*

Since the objective function J in (LS1) is the composition of a convex function $\mathrm{trace}\,(\bar{L}^{-1})$ of a positive definite matrix $\bar{L} \succ 0$ with an affine function $\bar{L} = L + D_\kappa D_x$, it follows that J is a convex function of x. By enlarging the Boolean constraint set $x_i \in \{0, 1\}$ to its convex hull $x_i \in [0, 1]$ (i.e., the smallest convex set that contains the Boolean constraint set), we obtain a



(a) (b)

Fig. 7: The degree distribution of (a) the random network of Section III-C.2 and of (b) 41 leaders selected using soft constraint method. Note that the soft constraint method chooses all nodes with degree less than 8 and all nodes with degree greater than 18.

convex relaxation of (LS1)

$$\begin{array}{ll} \underset{x}{\text{minimize}} & J(x) \;=\; \text{trace}\left((L \,+\, D_\kappa D_x)^{-1}\right) \\ \text{subject to} & \mathbb{1}^T x \;=\; N_l, \quad 0 \;\leq\; x_i \;\leq\; 1, \qquad i \;=\; 1, \ldots, n. \end{array} \tag{CR}$$

Since we have enlarged the constraint set, the solution x^* of the relaxed problem (CR) provides a lower bound on J_{opt}. However, x^* may not provide a selection of N_l leaders, as it may turn out not to be Boolean-valued. If x^* is Boolean-valued, then it is the global solution of (LS1).

Following similar argument as in Section III-A, Schur complement can be used to formulate the convex optimization problem (CR) as an SDP

$$\begin{array}{ll} \underset{X,\,x}{\text{minimize}} & \text{trace}\,(X) \\ \text{subject to} & \begin{bmatrix} X & I \\ I & L + D_\kappa D_x \end{bmatrix} \;\succeq\; 0 \\ & \mathbb{1}^T x \;=\; N_l, \quad 0 \;\leq\; x_i \;\leq\; 1, \quad i \;=\; 1, \ldots, n. \end{array}$$

For small networks (e.g., $n \leq 30$), this problem can be solved efficiently using standard SDP solvers. For large networks, we develop a customized interior point method in Appendix C.

B. Greedy algorithm to obtain an upper bound

With the lower bound on the optimal value $J_{\rm opt}$ resulting from the convex relaxation (CR) in Section IV-A, we next use a greedy algorithm to compute an upper bound on $J_{\rm opt}$. This algorithm selects one leader at a time by assigning the node that provides the largest performance improvement as the leader. Once this is done, an attempt to improve a selection of N_l leaders is made by checking possible swaps between the leaders and the followers. Similar greedy algorithms have been used in [18], [19] for noise-free leader selection problem. In the noise-corrupted problem, we show that substantial improvement in algorithmic complexity can be achieved by exploiting structure of the low-rank modifications to the Laplacian matrix.

1) One-leader-at-a-time algorithm: As the name suggests, we select one leader at a time by assigning the node that results in the largest performance improvement as the leader. To select the first leader, we compute

$$J_1^i \;=\; \mathrm{trace}\left((L \,+\, \kappa_i e_i e_i^T)^{-1}\right)$$

for $i = 1, \ldots, n$, and assign the node, say v_1, that achieves the minimum value of $\{J_1^i\}$. If two or more nodes provide the largest performance improvement, we select one of these nodes as a leader. After choosing s leaders, $v_1, \ldots, v_s$, we compute

$$\begin{aligned} J_{s+1}^i &= \mathrm{trace}\left((L_s \,+\, \kappa_i e_i e_i^T)^{-1}\right) \\ L_s &= L \,+\, \kappa_{v_1} e_{v_1} e_{v_1}^T \,+\, \cdots \,+\, \kappa_{v_s} e_{v_s} e_{v_s}^T \end{aligned}$$

for $i \notin \{v_1, \ldots, v_s\}$, and select node v_{s+1} that yields the minimum value of $\{J_{s+1}^i\}$. This procedure is repeated until all N_l leaders are selected.

Without exploiting structure, the above procedure requires $O(n^4 N_l)$ operations. On the other hand, the rank-1 update formula obtained from matrix inversion lemma

$$(L_s \,+\, \kappa_i e_i e_i^T)^{-1} \;=\; L_s^{-1} \,-\, \frac{L_s^{-1}\, \kappa_i e_i e_i^T\, L_s^{-1}}{1 \,+\, \kappa_i e_i^T L_s^{-1} e_i} \tag{12}$$

yields

$$J_{s+1}^i \;=\; \mathrm{trace}\,(L_s^{-1}) \,-\, \frac{\kappa_i \|(L_s^{-1})_i\|_2^2}{1 + \kappa_i (L_s^{-1})_{ii}}$$

where $(L_s^{-1})_i$ is the ith column of L_s^{-1} and $(L_s^{-1})_{ii}$ is the iith entry of L_s^{-1}. To initiate the

algorithm, we use the generalized rank-1 update [34],

$$L_1^{-1} = L^\dagger - (L^\dagger e_i)\mathbb{1}^T - \mathbb{1}(L^\dagger e_i)^T + ((1/\kappa_i) + e_i^T L^\dagger e_i)\mathbb{1}\mathbb{1}^T$$

and thus,

$$J_1^i = \operatorname{trace}(L^\dagger) + n((1/\kappa_i) + e_i^T L^\dagger e_i)$$

where $L^\dagger$ denotes the pseudo-inverse of L (e.g., see [35])

$$L^\dagger = (L + \mathbb{1}\mathbb{1}^T/n)^{-1} - \mathbb{1}\mathbb{1}^T/n.$$

Therefore, once L_s^{-1} is determined, the inverse of the matrix on the left-hand-side of (12) can be computed using $O(n^2)$ operations and J_{s+1}^i can be evaluated using $O(n)$ operations. Overall, N_l rank-1 updates, $nN_l/2$ objective function evaluations, and one full matrix inverse (for computing L_s^{-1}) require $O(n^2N_l + n^3)$ operations as opposed to $O(n^4N_l)$ operations without exploiting the low-rank structure. In large-scale networks, further computational advantage can be gained by exploiting structure of the underlying Laplacian matrices; see [36].

2) Swap algorithm: Having determined a selection of leaders using one-leader-at-a-time algorithm, we swap one of the N_l leaders with one of the $n - N_l$ followers, and check if such a swap leads to a decrease in J. If no decrease occurs for all $(n - N_l)N_l$ swaps, the algorithm terminates; if a decrease in J occurs, we update the leader and then restart checking the possible $(n - N_l)N_l$ swaps for the new leader selection. This swap procedure has been used as an effective means for improving performance of combinatorial algorithms encountered in graph partitioning [37], sensor selection [38], and community detection problems [39].

Since a swap between a leader i and a follower j leads to a rank-2 modification (13) to the matrix $\bar{L} = L + D_\kappa D_x$, we can exploit this low-rank structure to gain computational efficiency. Using the matrix inversion lemma, we have

$$\left(\bar{L} - \kappa_i e_i e_i^T + \kappa_j e_j e_j^T\right)^{-1} = \bar{L}^{-1} - \bar{L}^{-1}\bar{E}_{ij}(I_2 + E_{ij}^T\bar{L}^{-1}\bar{E}_{ij})^{-1}E_{ij}^T\bar{L}^{-1} \tag{13}$$

where $E_{ij} = [\,e_i \quad e_j\,]$, $\bar{E}_{ij} = [\,-\kappa_i e_i \quad \kappa_j e_j\,]$, and I_2 is the 2×2 identity matrix. Thus, the objective function after the swap between leader i and follower j is given by

$$J_{ij} = J - \operatorname{trace}\left((I_2 + E_{ij}^T\bar{L}^{-1}\bar{E}_{ij})^{-1}E_{ij}^T\bar{L}^{-2}\bar{E}_{ij}\right). \tag{14}$$

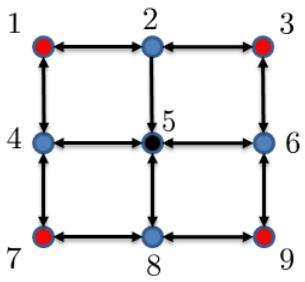


Fig. 8: A 3×3 grid.

Here, we do not need to form the full matrix $\bar{L}^{-2}$, since

$$E_{ij}^T \bar{L}^{-2} \bar{E}_{ij} = \begin{bmatrix} -\kappa_i (\bar{L}^{-2})_{ii} & \kappa_j (\bar{L}^{-2})_{ij} \\ -\kappa_i (\bar{L}^{-2})_{ji} & \kappa_j (\bar{L}^{-2})_{jj} \end{bmatrix}$$

and the ijth entry of $\bar{L}^{-2}$ can be computed by multiplying the ith row of $\bar{L}^{-1}$ with the jth column of $\bar{L}^{-1}$. Thus, evaluation of J_{ij} takes $O(n)$ operations and computation of the matrix inverse in (13) requires $O(n^2)$ operations.

Remark 1: Since the total number of swaps for large-scale networks can be large, we follow [38] and limit the maximum number of swaps with a linear function of the number of nodes n. On the other hand, particular structure of networks can be exploited to reduce the required number of swaps. To illustrate this, let us consider the problem of selecting one leader in a network with 9 nodes shown in Fig. 8. Suppose that nodes in set $S_1 := \{1, 3, 7, 9\}$ have the same feedback gain κ_1 and that nodes in set $S_2 := \{2, 4, 6, 8\}$ have the same feedback gain κ_2. In addition, suppose that node 5 is chosen as a leader. Owing to symmetry, to check if selecting other nodes as a leader can improve performance we only need to swap node 5 with one node in each set S_1 and S_2. We note that more sophisticated symmetry exploitation techniques have been discussed in [26], [40].

C. Alternating direction method of multipliers

Since the previously introduced greedy algorithm may not yield an optimal selection of leaders, we next employ the ADMM algorithm [17] as an alternative approach to a selection of N_l leaders for problem (LS1). Although the convergence of this method depends on the initial conditions and on the algorithmic parameters, ADMM is capable of handling the nonconvex Boolean constraints

explicitly by a simple projection onto a discrete nonconvex set

$$\mathcal{C} := \left\{ x \mid \mathbb{1}^T x = N_l,\ x_i \in \{0,1\},\ i = 1,\ldots,n \right\}. \tag{15}$$

We can rewrite (LS1) as an unconstrained optimization problem

$$\underset{X,\,x}{\text{minimize}} \quad J(x) + \mathcal{I}(x) \tag{16}$$

where $\mathcal{I}(x)$ is indicator function associated with set $\mathcal{C}$

$$\mathcal{I}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ +\infty & \text{if } x \notin \mathcal{C}. \end{cases}$$

Now, (16) can be put into the following equivalent form suitable for the application of ADMM

$$\begin{aligned} \underset{x,\,z}{\text{minimize}} \quad & J(x) + \mathcal{I}(z) \\ \text{subject to} \quad & x - z = 0 \end{aligned} \tag{17}$$

and the augmented Lagrangian associated with (17) is given by

$$\mathcal{L}_\rho(x,z,\lambda) = J(x) + \mathcal{I}(z) + \lambda^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2$$

where $\lambda \in \mathbb{R}^n$ is the dual variable and ρ is a positive number. For $k = 0, 1, \ldots$, the ADMM algorithm updates x, z, and λ in an iterative fashion

$$x^{k+1} := \arg\min_x \mathcal{L}_\rho(x, z^k, \lambda^k) \tag{18a}$$

$$z^{k+1} := \arg\min_z \mathcal{L}_\rho(x^{k+1}, z, \lambda^k) \tag{18b}$$

$$\lambda^{k+1} := \lambda^k + \rho(x^{k+1} - z^{k+1}) \tag{18c}$$

until $\|x^{k+1} - z^{k+1}\|_2 \le \epsilon$ and $\|z^{k+1} - z^k\|_2 \le \epsilon$.

Splitting the optimization variables into two copies $\{x, z\}$ and updating them in an alternating fashion yields the minimization problems (18a) and (18b) that are easy to solve.

1) x-minimization step: By completion of squares in $\mathcal{L}_\rho$ with respect to x, problem (18a) can be expressed as

$$\underset{x}{\text{minimize}} \quad \text{trace}\left((L + D_\kappa D_x)^{-1}\right) + \frac{\rho}{2}\|x - u^k\|_2^2 \tag{19}$$

where

$$u^k \,:=\, z^k \,-\, (1/\rho)\lambda^k.$$

Since (19) is equivalent to the following problem,

$$\begin{array}{ll} \underset{x,\,\mu}{\text{minimize}} & \text{trace}\,((L \,+\, D_\kappa D_x)^{-1}) \,+\, \mu \\ \text{subject to} & \dfrac{\rho}{2}\,\|x \,-\, u^k\|_2^2 \,\leq\, \mu \end{array}$$

it can be expressed as an SDP

$$\begin{array}{ll} \underset{X,\,x,\,\mu}{\text{minimize}} & \text{trace}\,(X) \,+\, \mu \\ \text{subject to} & \left[\begin{array}{cc} X & I \\ I & L + D_\kappa D_x \end{array}\right] \,\succeq\, 0 \\ & \left[\begin{array}{cc} I & x - u^k \\ (x - u^k)^T & 2\mu/\rho \end{array}\right] \,\succeq\, 0 \end{array}$$

where the second LMI, resulting from the use of Schur complement, is an alternative way of writing the quadratic constraint

$$2\mu/\rho \,-\, (x - u^k)^T(x - u^k) \,\geq\, 0.$$

Thus, for small networks, problem (19) can be solved efficiently using standard SDP solvers. For large networks, we use descent methods [32] (e.g., Newton's method) with the gradient and Hessian of $\mathcal{L}_\rho$ with respect to x being given by

$$\begin{aligned} \nabla\mathcal{L}_\rho &= -\,\kappa \circ \text{diag}\,((L + D_\kappa D_x)^{-2}) \,+\, \rho\,(x \,-\, u^k) \\ \nabla^2\mathcal{L}_\rho &= 2\,(D_\kappa(L + D_\kappa D_x)^{-2}D_\kappa) \circ (L + D_\kappa D_x)^{-1} \,+\, \rho I \end{aligned}$$

where $\text{diag}\,(M)$ denotes the vector determined by the main diagonal of a matrix M.

2) z-minimization step: Using similar argument as in [17, Section 9.1] (see Appendix D for details), the z-minimization problem (18b) can be solved explicitly using a simple projection onto the set $\mathcal{C}$

$$z_i \,=\, \begin{cases} 1 & \text{if } v_i^k \,\geq\, [v^k]_{N_l} \\ 0 & \text{if } v_i^k \,<\, [v^k]_{N_l} \end{cases} \tag{20}$$

TABLE II: Lower and upper bounds on the noise-corrupted leader selection problem (LS1) for the example shown in Fig. 1. Lower bounds $J_{\rm lb}$ are obtained by solving the convex relaxation (CR); upper bounds $J_{\rm ub}$ from greedy algorithm – the one-leader-at-a-time algorithm followed by the swap algorithm – are actually tight, i.e., $J_{\rm ub} = J_{\rm opt}$; upper bounds $J_{\rm ub}$ from ADMM are tight for $N_l = 4, 5$.

		greedy algorithm		ADMM	
N_l	$J_{\rm lb}$	$J_{\rm ub}$	leaders	$J_{\rm ub}$	leaders
1	38.4	72.3	13	118.3	25
2	30.3	43.4	8, 25	47.9	7, 25
3	26.7	35.2	8, 16, 25	36.7	7, 16, 25
4	24.3	30.0	3, 7, 16, 25	30.0	3, 7, 16, 25
5	22.4	25.8	3, 7, 9, 16, 25	25.8	3, 7, 9, 16, 25

where

$$v^k \,:=\, x^{k+1} \,+\, (1/\rho)\lambda^k$$

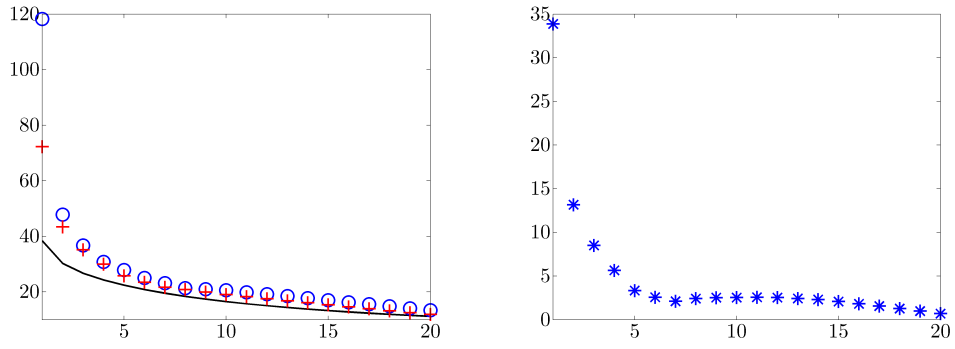
and $[v^k]_{N_l}$ denotes the (N_l)th largest entry of v^k. We note that reference [17] provides related projections onto several important nonconvex sets.

D. Examples

We next provide several examples to illustrate the performance of the developed methods. In all examples we set κ_i to be the degree of node i. We set the initial conditions of the ADMM algorithm to $\{z^0 = 0,\, \lambda^0 = 0\}$ and the penalty weight to $\rho = 10^3$.

1) A small network from [18]*:* For the example discussed in Section III-C.1 with $N_l \leq 5$, we determine the global minima to the noise-corrupted leader selection problem (LS1) by exhaustive search. It turns out that the one-leader-at-a-time algorithm followed by the swap algorithm actually finds the global minima. As shown in Table II, ADMM provides the global minima for the problems with 4 and 5 leaders. It is also worth mentioning that the globally optimal selection of noise-corrupted leaders coincides with the globally optimal selection of noise-free leaders (cf. Table I).

Figure 9a shows lower bounds resulting from convex relaxation and upper bounds resulting from ADMM and from greedy algorithm. As the number of leaders N_l increases, the gap between the lower and upper bounds from greedy algorithm decreases; see Fig. 9b.



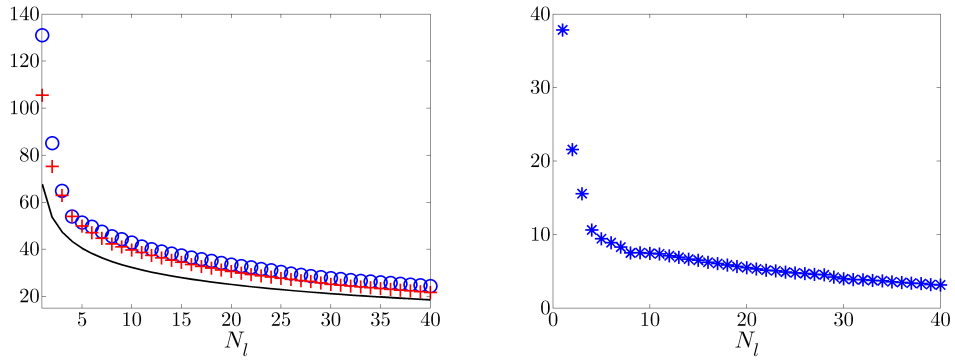
(a) (b)

Fig. 9: The network with 25 nodes: (a) lower bounds ($-$) resulting from convex relaxation and upper bounds resulting from greedy algorithm (i.e., one-leader-at-a-time algorithm followed by swap algorithm) ($+$) and from ADMM ($\circ$); (b) the gap between lower bounds and upper bounds resulting from greedy algorithm.

2) A 2D lattice: We next consider the leader selection problem for a 9×9 regular lattice. Figure 10a shows lower bounds resulting from convex relaxation and upper bounds resulting from ADMM and from greedy algorithm, i.e., the one-leader-at-a-time algorithm followed by the swap algorithm. As the number of leaders N_l increases, the gap between the lower and upper bounds from greedy algorithm decreases; see Fig. 10b. For $N_l = 1, \ldots, 40$, the number of swap updates ranges between 1 and 19 and the average number of swaps is 10.

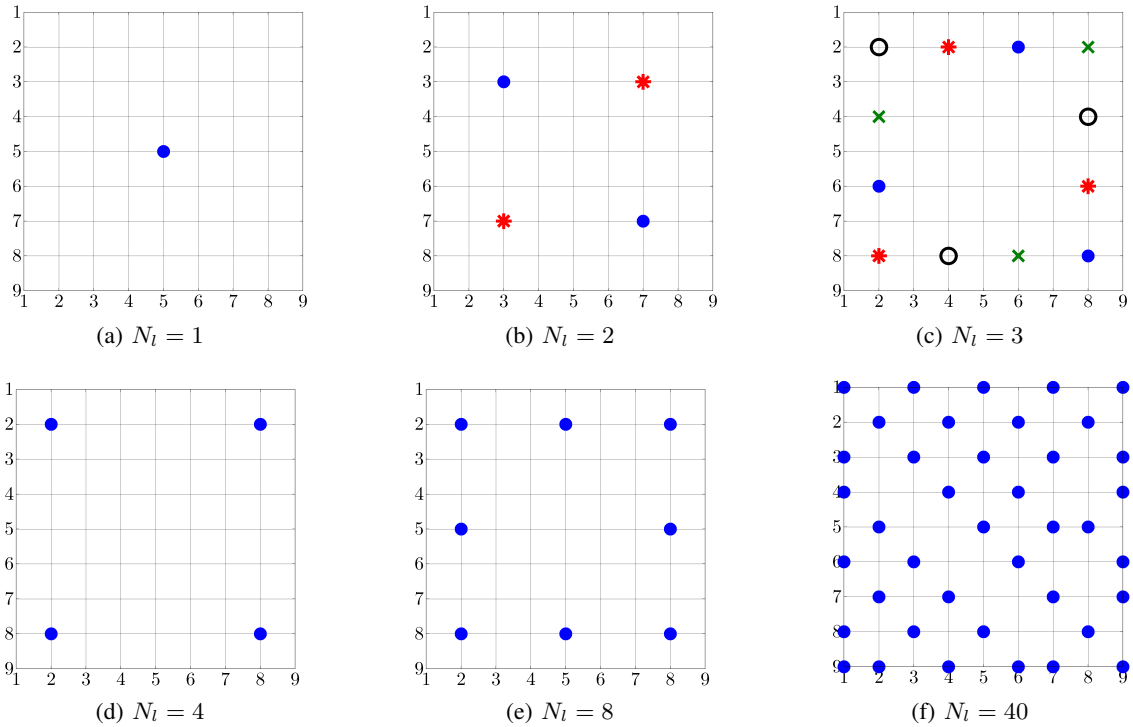
Figure 11 shows selection of leaders resulting from the greedy algorithm for different choices of N_l. For $N_l = 1$, the *center* node $(5, 5)$ provides the optimal selection of a single leader. As N_l increases, nodes away from the center node $(5, 5)$ are selected; for example, for $N_l = 2$, nodes $\{(3, 3), (7, 7)\}$ are selected and for $N_l = 3$, nodes $\{(2, 6), (6, 2), (8, 8)\}$ are selected. Selection of nodes farther away from the center becomes more significant for $N_l = 4$ and $N_l = 8$.

The selection of leaders exhibits symmetry shown in Fig. 11. In particular, when N_l is large, almost uniform spacing between the leaders is observed; see Fig. 11f for $N_l = 40$. This is in contrast to the selection of leaders along the boundary nodes in the random network example in Fig. 6c. For the random network example in Section III-C.2, the selection of the noise-corrupted leaders resembles that of the noise-free leaders (results are omitted for brevity).



(a) (b)

Fig. 10: A 2D lattice: (a) lower bounds ($-$) resulting from convex relaxation and upper bounds resulting from greedy algorithm (i.e., one-leader-at-a-time algorithm followed by swap algorithm) ($+$) and from ADMM ($\circ$); (b) the gap between lower bounds and upper bounds resulting from greedy algorithm.



(a) $N_l = 1$ (b) $N_l = 2$ (c) $N_l = 3$

(d) $N_l = 4$ (e) $N_l = 8$ (f) $N_l = 40$

Fig. 11: Selections of leaders ($\bullet$) obtained using the one-at-a-time algorithm followed by the swap algorithm for a 2D lattice. The two selections of two leaders denoted by ($\bullet$) and ($*$) in (b) provide the same objective function J. The four selections of three leaders denoted by ($\bullet$), ($*$), ($\times$), and ($\circ$) in (c) provide the same J.

V. CONCLUDING REMARKS

The main contribution of this paper is the development of efficient algorithms that facilitate selection of leaders in large stochastically forced consensus networks. For the noise-corrupted leader selection problem (LS1), we focus on computing lower and upper bounds on the global optimal value. A lower bound is obtained by solving a convex relaxation, and upper bounds result from a simple but efficient greedy algorithm and the alternating direction method of multipliers. For the noise-free leader selection problem (LS2), we provide an explicit expression for the variance amplification of the network. This allows us to identify sources of nonconvexity and to propose a convex relaxation of the objective function in (LS2). Furthermore, we use augmentation of the objective function with the ℓ_1 norm of the vector of optimization variables as a surrogate for obtaining a sparse solution whose nonzero elements identify the leaders. Several examples are provided to illustrate the effectiveness of our algorithms. We are currently applying these tools for leader selection problems in different types of networks, including small-world social networks.

APPENDIX

A. Equivalence between leader selection and sensor selection problems

We next show that the problem of choosing N_l absolute position measurements among n sensors to minimize the variance of the estimation error in Section II-B is equivalent to the noise-corrupted leader selection problem (LS1).

Given the measurement vector y in (5), the linear minimum variance unbiased estimate of ψ is determined by [41, Chapter 4.4]

$$\hat{\psi} \,=\, (E_r W_r^{-1} E_r^T + E_a (E_a^T W_a E_a)^{-1} E_a^T)^{-1} (E_r W_r^{-1} y_r + E_a (E_a^T W_a E_a)^{-1} y_a)$$

with the covariance of the estimation error

$$\Sigma \,=\, \mathcal{E}((\psi - \hat{\psi})(\psi - \hat{\psi})^T) \,=\, (E_r W_r^{-1} E_r^T + E_a (E_a^T W_a E_a)^{-1} E_a^T)^{-1}.$$

Furthermore, let us assume that

$$W_r \,=\, I, \quad W_a \,=\, D_\kappa^{-1}.$$

The choice of W_a indicates that a larger value of κ_i corresponds to a more accurate absolute

measurement of sensor i. Then

$$(E_a^T W_a E_a)^{-1} \;=\; (E_a^T D_\kappa^{-1} E_a)^{-1} \;=\; E_a^T D_\kappa E_a$$

and thus,

$$\Sigma \;=\; (E_r E_r^T + E_a E_a^T D_\kappa E_a E_a^T)^{-1}.$$

Since $E_a E_a^T$ is a diagonal matrix with its ith diagonal element being 1 for $i \in \mathcal{I}_a$ and $E_r E_r^T$ is the Laplacian matrix of the relative measurement graph, it follows that

$$D_x \;=\; E_a E_a^T, \quad L \;=\; E_r E_r^T, \quad \Sigma \;=\; (L \;+\; D_x D_\kappa D_x)^{-1} \;=\; (L \;+\; D_\kappa D_x)^{-1}$$

where $D_x D_\kappa D_x = D_\kappa D_x$ because D_x and D_κ commute and $D_x D_x = D_x$. Therefore, we have established equivalence between the noise-corrupted leader selection problem (LS1) and the problem of choosing N_l sensors with absolute position measurements such that the variance of the estimation error is minimized.

To formulate an estimation problem that is equivalent to the noise-free leader selection problem (LS2), we follow [10] and assume that the positions of N_l sensors are known *a priori*. Let ψ_l denote the positions of these *reference sensors* and let ψ_f denote the positions of the other sensors. We can thus write the relative measurement equation (4) as

$$y_r \;=\; E_r^T \psi \;+\; w_r \;=\; E_l^T \psi_l \;+\; E_f^T \psi_f \;+\; w_r$$

and the linear minimum variance unbiased estimate of ψ_f is given by

$$\hat{\psi}_f \;=\; (E_f E_f^T)^{-1} E_f W_r^{-1} \,(y_r \;-\; E_l^T \psi_l)$$

with covariance of the estimation error

$$\Sigma_f \;=\; (E_f E_f^T)^{-1}.$$

Identifying $E_f E_f^T$ with L_f in the Laplacian matrix

$$L \;=\; E_r E_r^T \;=\; \begin{bmatrix} E_l E_l^T & E_l E_f^T \\ E_f E_l^T & E_f E_f^T \end{bmatrix} \;=\; \begin{bmatrix} L_l & L_0 \\ L_0^T & L_f \end{bmatrix}$$

establishes equivalence between problem (LS2) and the problem of assigning N_l sensors with known reference positions to minimize the variance of the estimation error of sensor network.

B. ADMM for the soft constraint method

We next employ ADMM for the soft constraint method developed in Section III-B. We consider the following minimization problem

$$\underset{x}{\text{minimize}} \quad f(x) + \gamma \|x\|_{\ell_1}$$

where f is the convex approximation of (6)

$$f(x) = \text{trace}\left((I - D_x)(G_0 + D_x \circ L)^{-1}(I - D_x)\right)$$

and G_0 is the linear approximation of G given by (10). This problem is equivalent to the constrained problem

$$\begin{aligned} \underset{x,\, z}{\text{minimize}} \quad & f(x) + \gamma \|z\|_{\ell_1} \\ \text{subject to} \quad & x - z = 0 \end{aligned}$$

and the associated augmented Lagrangian function is given by

$$\mathcal{L}_\rho(x, z, \lambda) = f(x) + \gamma \|z\|_{\ell_1} + \lambda^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2.$$

By completion of squares in $\mathcal{L}_\rho$ with respect to z, the z-minimization problem (18b) can be expressed as

$$\underset{z}{\text{minimize}} \quad \gamma \|z\|_{\ell_1} + \frac{\rho}{2}\|z - v^k\|_2^2$$

where $v^k = x^{k+1} + (1/\rho)\lambda^k$. The solution is given by the soft thresholding operator (e.g., see [17, Section 4.4.3])

$$z_i^* = \mathcal{S}_{\gamma/\rho}(v_i^k) = \begin{cases} \left(1 - \dfrac{\gamma/\rho}{|v_i^k|}\right) v_i^k, & |v_i^k| > \gamma/\rho \\ 0, & |v_i^k| \le \gamma/\rho \end{cases} \tag{21}$$

for $i = 1, \ldots, n$. On the other hand, by completing squares in $\mathcal{L}_\rho$ with respect to x, we obtain

$$\underset{x}{\text{minimize}} \quad \phi(x) = f(x) + \frac{\rho}{2}\|x - u^k\|_2^2$$

where $u^k = z^k - (1/\rho)\lambda^k$. This problem can be solved using descent methods (e.g., gradient method [32]). Here, we provide the expression for the gradient of ϕ

$$\begin{aligned}\nabla\phi(x) \;\; = \;\; & -2\,\mathrm{diag}\,((I-D_x)M^{-1}) \,+\, \mathrm{diag}\,(L(I-D_{x_0})M^{-1}(I-D_x)^2M^{-1}) \\ & -\mathrm{diag}\,(M^{-1}(I-D_x)^2M^{-1})\circ\mathrm{diag}\,(L) \,+\, \rho(x-u^k)\end{aligned}$$

where $M = G_0 + D_x \circ L$.

C. Customized interior point method for (CR)

We begin by augmenting the objective function in (CR) with log-barrier functions associated with the inequality constraints on x_i

$$\begin{aligned}&\underset{x}{\text{minimize}} \quad q(x) \,=\, \tau\,\mathrm{trace}\left((L \,+\, D_\kappa D_x)^{-1}\right) \,+\, \sum_{i=1}^{n}\big(-\,\log(x_i) \,-\, \log(1-x_i)\big) \\ &\text{subject to} \quad \mathbb{1}^T x \,=\, N_l.\end{aligned} \tag{22}$$

The solution of the approximate problems (22) converges to the solution of the convex relaxation (CR) as the positive scalar τ increases to infinity [32, Section 11.2]. We solve a sequence of problem (22) by gradually increasing τ, and by starting each minimization using the solution from the previous value of τ. We use Newton's method to solve (22) for a fixed τ, and the Newton direction for problems with linear constraints is given by (e.g., see [32, Section 10.2])

$$x_{\mathrm{nt}} \,=\, -\,(\nabla^2 q)^{-1}\nabla q \,-\, \delta(\nabla^2 q)^{-1}\mathbb{1}$$

where

$$\delta \,=\, -\frac{\mathbb{1}^T(\nabla^2 q)^{-1}\nabla q}{\mathbb{1}^T(\nabla^2 q)^{-1}\mathbb{1}}.$$

Here, the expressions for the ith entry of the gradient direction ∇q and for the Hessian matrix are given by

$$\begin{aligned}(\nabla q)_i \;\; &= \;-\; \tau\,\kappa_i\,((L+D_\kappa D_x)^{-2})_{ii} \,-\, x_i^{-1} \,-\, (x_i-1)^{-1} \\ \nabla^2 q \;\; &= 2\tau\,(D_\kappa(L+D_\kappa D_x)^{-2}D_\kappa)\circ(L+D_\kappa D_x)^{-1} \,+\, \mathrm{diag}\left(x_i^{-2} \,+\, (1-x_i)^{-2}\right).\end{aligned}$$

We next examine complexity of computing the Newton direction x_{nt}. The major cost of computing $\nabla^2 q$ is to form $(L+D_\kappa D_x)^{-2}$, which takes $(7/3)n^3$ operations to form $(L+D_\kappa D_x)^{-1}$

and n^3 operations to form $(L + D_\kappa D_x)^{-2}$. Computing $x_{\rm nt}$ requires solving two linear equations,

$$(\nabla^2 q)\, y \;=\; -\nabla q, \quad (\nabla^2 q)\, z \;=\; -\mathbb{1}$$

which takes $(1/3)n^3$ operations using Cholesky factorization. Thus, computation of each Newton step requires $(7/3 + 1 + 1/3)n^3 = (11/3)n^3$ operations.

D. Derivation of (20)

We use completion of squares to obtain the following problem which is equivalent to (18b)

$$\begin{array}{ll} \underset{z}{\text{minimize}} & (\rho/2)\|z \,-\, v^k\|_2^2 \\ \text{subject to} & z \,\in\, \mathcal{C} \end{array}$$

where $v^k = x^{k+1} + (1/\rho)\lambda^k$ and the set $\mathcal{C}$ is given by (15). Projecting v onto $\mathcal{C}$ yields

$$z_i \;=\; \begin{cases} 1 & \text{if} \;\; v_i^k \;\geq\; [v^k]_{N_l} \\ 0 & \text{if} \;\; v_i^k \;<\; [v^k]_{N_l} \end{cases} \tag{23}$$

where $[v^k]_{N_l}$ is the (N_l)th largest entry of v^k. To see this, consider $\bar{z} \in \mathcal{C}$, i.e., $\mathbb{1}^T\bar{z} = N_l$ and $\bar{z}_i \in \{0, 1\}$, but $\bar{z}$ is not the projection determined by (23). Thus, there exists at least one entry of $\bar{z}$, say the rth entry, such that $\bar{z}_r = 1$ for $v_r^k < [v^k]_{N_l}$, and at least one entry, say the jth entry, such that $\bar{z}_j = 0$ for $v_j^k \geq [v^k]_{N_l}$. Consider

$$\delta_{rj} \;=\; (\bar{z}_r - v_r^k)^2 \,+\, (\bar{z}_j - v_j^k)^2 \;=\; (1 - v_r^k)^2 \,+\, (v_j^k)^2$$

and $\delta_{jr} = (v_r^k)^2 + (1 - v_j^k)^2$. Since $\delta_{rj} - \delta_{jr} = 2(v_j^k - v_r^k) > 0$, it follows that the objective function $(\rho/2)\|z - v^k\|_2^2$ will decrease if we choose $\{\bar{z}_r = 0,\, \bar{z}_j = 1\}$ instead of $\{\bar{z}_r = 1,\, \bar{z}_j = 0\}$. Therefore, we can reduce the objective function by exchanging the values of two entries $\bar{z}_r = 1$ (with $v_r^k < [v^k]_{N_l}$) and $\bar{z}_j = 0$ (with $v_j^k \geq [v^k]_{N_l}$) until (23) is satisfied for all $i = 1, \ldots, n$.

REFERENCES

[1] M. Mesbahi and M. Egerstedt, *Graph-theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[2] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.

[3] B. Golub and M. Jackson, "Naive learning social networks and the wisdom of crowds," *American Economic Journal: Microeconomics*, vol. 2, no. 1, pp. 112–149, 2010.

[4] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *J. Parallel Distrib. Comput.*, vol. 7, no. 2, pp. 279–301, 1989.

[5] J. E. Boillat, "Load balancing and Poisson equation in a graph," *Concurrency: Practice and Experience*, vol. 2, no. 4, pp. 289–313, 1990.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[7] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[8] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Automat. Control*, vol. 50, no. 2, pp. 169–182, 2005.

[9] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2007.

[10] P. Barooah and J. P. Hespanha, "Estimation on graphs from relative measurements: Distributed algorithms and fundamental limits," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 57–74, 2007.

[11] P. Barooah and J. P. Hespanha, "Estimation from relative measurements: Electrical analogy and large graphs," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2181–2193, 2008.

[12] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, 2007.

[13] G. F. Young, L. Scardovi, and N. E. Leonard, "Robustness of noisy consensus dynamics with directed communication," in *Proceedings of the 2010 American Control Conference*, 2010, pp. 6312–6317.

[14] D. Zelazo and M. Mesbahi, "Edge agreement: Graph-theoretic performance bounds and passivity analysis," *IEEE Trans. Automat. Control*, vol. 56, no. 3, pp. 544–555, 2011.

[15] B. Bamieh, M. R. Jovanović, P. Mitra, and S. Patterson, "Coherence in large-scale networks: dimension dependent limitations of local feedback," *IEEE Trans. Automat. Control*, vol. 57, no. 9, pp. 2235–2249, September 2012.

[16] F. Lin, M. Fardad, and M. R. Jovanović, "Optimal control of vehicular formations with nearest neighbor interactions," *IEEE Trans. Automat. Control*, vol. 57, no. 9, pp. 2203–2218, September 2012.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[18] S. Patterson and B. Bamieh, "Leader selection for optimal network coherence," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 2692–2697.

[19] A. Clark and R. Poovendran, "A submodular optimization framework for leader selection in linear multi-agent systems," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 3614–3621.

[20] A. Clark, L. Bushnell, and R. Poovendran, "A supermodular optimization framework for leader selection under link noise in linear multi-agent systems," *IEEE Trans. Automat. Control*, 2012, submitted; also arXiv:1208.0946v1.

[21] H. Kawashima and M. Egerstedt, "Leader selection via the manipulability of leader-follower networks," in *Proceedings of the 2012 American Control Conference*, 2012, pp. 6053–6058.

[22] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[23] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl*, vol. 14, pp. 877–905, 2008.

[24] H. G. Tanner, "On the controllability of nearest neighbor interconnections," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, 2004, pp. 2467–2472.

[25] B. Liu, T. Chu, L. Wang, and G. Xie, "Controllability of a leader-follower dynamic network with switching topology," *IEEE Trans. Automat. Control*, vol. 53, no. 4, pp. 1009–1013, 2008.

[26] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph theoretic perspective," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 162–186, 2009.

[27] Z. Jia, Z. Wang, H. Lin, and Z. Wang, "Interconnection topologies for multi-agent coordination under leader-follower framework," *Automatica*, vol. 45, no. 12, pp. 2857–2863, 2009.

[28] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 6605–6611.

[29] D. Zelazo, S. Schuler, and F. Allgöwer, "Performance and design of cycles in consensus networks," *Syst. Control Lett.*, 2012, submitted.

[30] D. Zelazo, S. Schuler, and F. Allgöwer, "Cycles and sparse design of consensus networks," in *Proceedings of the 51st IEEE Conference on Decision and Control*, 2012, to appear.

[31] L. El Ghaoui, F. Oustry, , and M. AitRami, "A cone complementarity linearization algorithm for static output-feedback and related problems," *IEEE Trans. Automat. Control*, vol. 42, no. 8, pp. 1171–1176, 1997.

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[33] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/, 2011.

[34] C. D. Meyer, "Generalized inversion of modified matrices," *SIAM Journal of Applied Mathematics*, vol. 24, no. 3, pp. 315–323, 1973.

[35] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Review*, vol. 50, no. 1, pp. 37–66, 2008.

[36] D. A. Spielman, "Algorithms, graph theory, and linear equations in Laplacian matrices," *Proceedings of the International Congress of Mathematicians*, vol. IV, pp. 2698–2722, 2010.

[37] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, pp. 291–307, 1970.

[38] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, 2009.

[39] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, p. 036104, 2006.

[40] S. Boyd, P. Diaconis, P. Parrilo, and L. Xiao, "Fastest mixing Markov chain on graphs with symmetries," *SIAM J. Optim.*, vol. 20, no. 2, pp. 792–819, 2009.

[41] D. G. Luenberger, *Optimization by Vector Space Methods*. John Wiley & Sons, 1968.